IN THE CLAIMS

1. (Currently Amended) A method of carrying packetized voice data over a data network, the method comprising:

aggregating multiple datagrams bound for a common intermediate switching point into a tunnel packet payload, each datagram comprising a voice data payload from a selected voice data stream and a compressed header-formatted header from which—when combined with information from one or more previous datagram headers from the same voice data stream—the datagram's destination endpoint can be derived;

encapsulating the tunnel packet payload in a tunnel packet having a tunnel header and forwarding the tunnel packet to the intermediate switching point; and

deaggregating the tunnel packet payload at the intermediate switching point and assigning the datagrams to new tunnel packet payloads based on a destination endpoint derived from each datagram's header and one or more previous datagram headers from the same voice data stream[[.]];

wherein the tunnel header is configured so that a packet-switched network device routing or forwarding packets according to a network address does not reference the tunnel packet payload.

2. (Original) The method of claim 1, further comprising, at the intermediate switching point, encapsulating each of the new tunnel packet payloads in a corresponding tunnel packet addressed either to the destination endpoint indicated for that tunnel packet's payload datagrams, or to a second intermediate switching point.

3. (Original) The method of claim 2, wherein each tunnel packet is encapsulated in an Internet Protocol datagram for network transport.

4. (Original) The method of claim 2, wherein assigning the datagrams to new tunnel packet payloads comprises:

matching each datagram's header information to a switching context maintained by the intermediate switching point, the switching context updated based on datagram header information for datagrams matched to that switching context; and

placing the datagram in a new tunnel packet payload selected based on the destination endpoint indicated in the matched switching context.

5. (Previously Presented) A method of carrying packetized voice data over a data network, the method comprising:

aggregating multiple datagrams bound for a common intermediate switching point into a tunnel packet payload, each datagram comprising a voice data payload from a selected voice data stream and a compressed header-formatted header from which—when combined with information from one or more previous datagram headers from the same voice data stream—the datagram's destination endpoint can be derived;

encapsulating the tunnel packet payload in a tunnel packet and forwarding the tunnel packet to the intermediate switching point;

deaggregating the tunnel packet payload at the intermediate switching point and assigning the datagrams to new tunnel packet payloads based on a destination endpoint derived from each datagram's header and one or more previous datagram headers from the same voice data stream; and

occasionally resetting the switching context by sending a datagram for that switching context with an uncompressed header.

6. (Previously Presented) A method of carrying packetized voice data over a data network, the method comprising:

aggregating multiple datagrams bound for a common intermediate switching point into a tunnel packet payload, each datagram comprising a voice data payload from a selected voice data stream and a compressed header-formatted header from which—when combined with information from one or more previous datagram headers from the same voice data stream—the datagram's destination endpoint can be derived;

encapsulating the tunnel packet payload in a tunnel packet and forwarding the tunnel packet to the intermediate switching point;

deaggregating the tunnel packet payload at the intermediate switching point and assigning the datagrams to new tunnel packet payloads based on a destination endpoint derived from each datagram's header and one or more previous datagram headers from the same voice data stream; and

resetting the switching context to a new voice data stream by sending a datagram for that switching context with an uncompressed header.

7. (Original) The method of claim 1, wherein each tunnel packet is encapsulated in one or more Asynchronous Transfer Mode cells for network transport.

8. (Currently Amended) A method of switching voice data within a data network, the method comprising:

receiving inbound tunnel packets, each inbound tunnel packet having the capability to carry a payload comprising multiple tunneled datagrams, each tunneled datagram having a corresponding header with a compressed header format that identifies it as belonging to a specified switching context;

parsing the payload from an inbound tunnel packet into individual incoming tunneled datagrams;

associating individual incoming tunneled datagrams with corresponding switching contexts;

mapping associated tunneled datagrams to outbound multiplexes based on the destination endpoint indicated in each datagram's associated context state;

updating the header for a tunneled datagram mapped to an outbound multiplex to identify that datagram with a switching context known to the outbound multiplex's destination; ~~and~~

aggregating tunneled datagrams assigned to a common outbound multiplex into an outbound tunnel packet payload; and

encapsulating the outbound tunnel packet payload using the Layer 2 Tunneling Protocol (L2TP);

wherein the corresponding headers are formatted by compressing voice data headers.

9. (Original) The method of claim 8, further comprising, after associating, decompressing the header of a tunneled datagram, and, prior to aggregating, compressing the header of the tunneled datagram.

10. (Currently Amended) A tunnel packet payload switch, wherein each tunnel packet has the capability to carry a payload comprising multiple tunneled datagrams, each tunneled datagram having a corresponding header with a compressed header format that identifies it as belonging to a specified switching context, the switch comprising:

a multiplex terminator to receive an inbound tunnel packet payload and parse the payload into individual incoming tunneled datagrams;

a context memory to store context state for multiple switching contexts;

a context matcher to associate individual incoming tunneled datagrams with corresponding switching contexts stored in the context memory;

a forwarding engine to map the associated tunneled datagrams to outbound multiplexes based on the destination endpoint indicated in each datagram's associated context state;

a header updater to revise the header for a tunneled datagram mapped to an outbound multiplex to identify that datagram with a switching context known to the outbound multiplex's destination;

a multiplexer to aggregate tunneled datagrams assigned to a common outbound multiplex into an outbound tunnel packet payload; and

a tunneler to datagram-encapsulate the outbound tunnel packet payload into a tunnel datagram using a layer two tunneling protocol[[.]];

wherein the revised header is configured so that a device routing or forwarding packets according at layer three can not examine contents of the outbound tunnel packet payload.

11. (Cancelled)

12. (Original) The switch of claim 10, wherein the inbound tunnel packet payload arrives at the switch encapsulated in a tunnel packet having a tunnel packet header that identifies the tunnel packet with an inbound multiplex, the multiplex terminator stripping the tunnel packet header and providing an indication of the tunnel packet's inbound multiplex to the context matcher.

13. (Original) The switch of claim 10, further comprising a decompressor to decompress compressed tunneled datagram headers, wherein the context matcher uses the decompressor output to update a full uncompressed header state in the context memory.

14. (Currently Amended) A tunnel packet payload switch, wherein each tunnel packet has the capability to carry a payload comprising multiple tunneled datagrams, each tunneled datagram having a corresponding header with a compressed header format that identifies it as belonging to a specified switching context, the switch comprising:

a multiplex terminator to receive an inbound tunnel packet payload and parse the payload into individual incoming tunneled datagrams;

a context memory to store context state for multiple switching contexts;

a context matcher to associate individual incoming tunneled datagrams with corresponding switching contexts stored in the context memory;

a forwarding engine to map the associated tunneled datagrams to outbound multiplexes based on the destination endpoint indicated in each datagram's associated context state;

a header updater to revise the header for a tunneled datagram mapped to an outbound multiplex to identify that datagram with a switching context known to the outbound multiplex's destination; and

a multiplexer to aggregate tunneled datagrams assigned to a common outbound multiplex into an outbound tunnel packet payload;

wherein the forwarding engine has the capability to reroute tunneled datagrams associated with a given switching context by changing a field in the switching context from one outbound multiplex identifier to another[[.]];

wherein the revised header is configured so that a router operating in a packet-switched network does not access the outbound tunnel packet payload.

15. (Original) The switch of claim 10, further comprising a timer in communication with the multiplexer, the multiplexer using the timer to dispatch an outbound tunnel packet payload when the first tunneled datagram to be assigned to a payload has been delayed by a maximum desired delay.

16. (Original) The switch of claim 15, the multiplexer also dispatching an outbound tunnel packet payload when the maximum desired delay has not been reached, but a desired payload size has been reached.

17. (Original) The switch of claim 10, the context matcher having the capability to construct a switching context for an incoming non-tunneled datagram and associate the incoming non-tunneled datagram with that switching context, the forwarding engine having the capability to map the associated non-tunneled datagram to an outbound multiplex.

18. (Original) The switch of claim 10, the forwarding engine having the capability to map an associated inbound tunneled datagram into an outbound non-tunneled datagram.

19. (Currently Amended) A tunnel packet payload switch, wherein each tunnel packet has the capability to carry a payload comprising multiple tunneled datagrams, each tunneled datagram having a corresponding header with a compressed header format that identifies it as belonging to a specified switching context, the switch comprising:

means for receiving inbound tunnel packet payloads and parsing the payloads into individual incoming tunneled datagrams;

means for associating individual incoming tunneled datagrams with corresponding switching contexts;

means for mapping associated tunneled datagrams to outbound multiplexes based on the destination endpoint indicated in each datagram's associated context state;

means for revising the header for a tunneled datagram mapped to an outbound multiplex to identify that datagram with a switching context known to the outbound multiplex's destination; and

means for aggregating tunneled datagrams assigned to a common outbound multiplex into an outbound tunnel packet payload; and

means for generating a tunnel header for encapsulating the outbound tunnel packet payload[[.]];

wherein the tunnel header is adapted to cause the outbound tunnel packet payload to tunnel under a layer three network layer.

20. (Currently Amended) An article of manufacture comprising a computer-readable medium containing a tunneled datagram processor program, the tunneled datagram processor program when executed causing a processor or several communicating processors to execute:

a context memory manager to update and manage a context memory that retains context state for multiple switching contexts, the context state for a given switching context including the destination endpoint for datagrams belonging to that context;

a context matcher to associate individual incoming tunneled datagrams with corresponding switching contexts stored in the context memory, wherein a tunneled datagram comprises a compressed-format header facilitating matching to a switching context;

a forwarder to map the associated tunneled datagrams to outbound multiplexes based on the destination endpoint indicated in each datagram's associated context state;

a header updater to revise the header for a tunneled datagram assigned to an outbound multiplex to identify that datagram with a switching context known to the outbound multiplex's destination; and

a multiplexer to aggregate tunneled datagrams assigned to a common outbound multiplex into an outbound tunnel packet payload;

wherein the compressed-formatted header is formed by compressing voice data headers and may be decompressed to form decompressed headers[[.]];

wherein the revised header causes the outbound tunnel packet payload to tunnel through a router that performs network layer routing.

21. (Original) The article of manufacture of claim 20, wherein the tunneled datagram processor program when executed also causes the processor or several communicating processors to execute:

a multiplex terminator to receive an inbound tunnel packet payload and parse the payload into the individual incoming tunneled datagrams.

22. (Original) The article of manufacture of claim 20, wherein a tunneled datagram further comprises a payload of compressed voice data.

23. (Previously Presented) The method of claim 1, wherein encapsulating further includes encapsulating using the Layer 2 Tunneling Protocol (L2TP).

24. (Cancelled)

25. (Previously Presented) The method of claim 1, wherein the compressed header-formatted headers are formatted by compressing an Internet Protocol (IP) header, a User Datagram Protocol (UDP) header, and a Real-time Transport Protocol (RTP) header.

26. (Previously Presented) The method of claim 1, wherein the compressed header-formatted headers are formatted by compressing a voice data header.